HN

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/839,527	04/20/2001	Dietrich Charisius	30013630-0013	2145

4678 7590 02/10/2005

MACCORD MASON PLLC
300 N. GREENE STREET, SUITE 1600
P. O. BOX 2974
GREENSBORO, NC 27402

EXAMINER	
SHRADER, LAWRENCE J	
ART UNIT	**PAPER NUMBER**
2124	

DATE MAILED: 02/10/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/839,527	CHARISIUS ET AL.	
	Examiner	Art Unit	
	Lawrence Shrader	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 September 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-128* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-128* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *6/25/2004*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed on 9/24/2004.

2. Claims 1 – 108 remain rejected; and new claims 109 – 128, which are added as requested by the Applicant, are also rejected.

Information Disclosure Statement

3. The missing reference "Object-Oriented Software Engineering, A Use Case Driven Approach," 1996, Jacobson, has been received, therefore all the information disclosure statements (IDS) submitted on 7/16/2001, 6/18/2002, and 8/13/2001 are acknowledged and have been considered.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1 – 3; 6 – 8, 11 – 14; 17 – 20; 23 – 26; 29 – 31; 34 – 37; 40 – 42; 45 – 47; 50 – 53; 56 – 59; 62 – 65; 68 – 70; 73 – 76; 79 – 82; 85 – 88; 91 – 94; 97 – 99; 102 – 105; and 108 – 128 are rejected under 35 U.S.C. 103(a) as being unpatentable over by Pazel, U.S. Patent

5,410,648 (Reissue 36,422) in view of Zgarba et al, U.S. Patent 6,502,239 (hereinafter referred to as Zgarba).

In regard to claim 1:

"displaying simultaneously a graphical and a textual representation of the source code,

wherein the graphical and the textual representations of the source code is generated from a language-neutral representation of the source code stored in a non-repository transient meta model, and

wherein the graphical representation has portions that correspond to the lines;

See Pazel Figures 6 and 7 with the corresponding text for simultaneously (column 5, lines 34 – 41) displaying graphical and textual representation of code. Pazel does not explicitly disclose textual representation generated from a language-neutral representation stored in a non-repository transient meta model. However, Zgarba discloses the textual representation generated from a language-neutral representation stored in a non-repository transient meta model (column 5, lines 1 – 45). The model need not be in a database (a repository) format as described at column 5, lines 18 - 19. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the simultaneous display of text and graphical of source code as taught by Pazel with the generation of graphical and text information from a non-repository transient meta model as taught by Zbarba because one would be motivated to generate updated code having immediate changes to specific parts of the software while the remaining code is untouched as taught by Zgarba in the Abstract.

"initiating an automated process that processes each of the lines; and

while the automated process processes each of the lines, displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated."

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 2, incorporating the rejection of claim 1:

> *"...further comprising the step of compiling the line before displaying the portion of the graphical representation that corresponds to the line."*

The debug program of Pazel is capable of disassembling executable code that has been previously compiled (column 3, lines 60 – 66).

In regard to claim 3, incorporating the rejection of claim 1:

> *"...wherein while the automated process processes each of the lines, the method further comprises the step of displaying the line of source code in a visually distinctive manner."*

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 109, incorporating the rejection of claim 1:

> *"...wherein the textual representation of the source code is obtained from the source code directly."*

See Pazel column 5, lines 42 – 49.

Claims 40 – 42, and 116 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 1 – 3, and 109 (corresponding methods).

Claim 108 and 128 (system) is rejected for the same corresponding reasons put forth in the rejection of claim 1, and 109 (corresponding method).

In regard to claim 6:

"..."displaying simultaneously a graphical and a textual representation of the source code,

wherein the graphical and the textual representations of the source code is generated from a language-neutral representation of the source code stored in a non-repository transient meta model, and

wherein the graphical representation has portions that correspond to the lines;

See Pazel Figures 6 and 7 with the corresponding text for simultaneously displaying graphical and textual representation of code. Pazel does not explicitly disclose textual representation generated from a language-neutral representation stored in a non-repository transient meta model. However, Zgarba discloses the textual representation generated from a language-neutral representation stored in a non-repository transient meta model (column 5, lines 1 – 45). The model need not be in a database (a repository) format as described at column 5, lines 18 - 19. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the simultaneous display of text and graphical of source code as taught by Pazel with the generation of graphical and text information from a non-repository transient meta model as taught by Zbarba because one would be motivated to generate updated code having immediate changes to specific parts of the software while the remaining code is untouched as taught by Zgarba in the Abstract.

for each of the lines, displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it appears that progression through the code is animated."

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 7, incorporating the rejection of claim 6:

"...further comprising the step of compiling the line before displaying the portion of the graphical representation that corresponds to the line."

The debug program of Pazel is capable of disassembling executable code that has been previously compiled (column 3, lines 60 – 66).

In regard to claim 8, incorporating the rejection of claim 6:

"...wherein for each of the lines, the method further comprises the step of displaying the line of source code in a visually distinctive manner."

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 110, incorporating the rejection of claim 6:

"...wherein the textual representation of the source code is obtained from the source code directly."

See Pazel column 5, lines 42 – 49.

Claims 45 – 47, and 117 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 6 – 8, and 110 (corresponding methods).

In regard to claim 11:

"...displaying a graphical representation of the plurality of lines such that at least one of the lines is not represented in the graphical representation;

initiating an automated process on each of the lines of the source code;"

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39). Figures 3 and 6 indicate that at least one of the lines is not represented in the graphical representation.

"receiving an indication to suspend the automated process when the automated process encounters one of the lines that is represented in the graphical representation; and

while the automated process is being performed on each of the lines of source code, determining whether the line is represented in the graphical representation; and

when it is determined that the line is represented in the graphical representation, suspending the automated process."

Pazel discloses a debug action in Figure 3. When it is determined that a subroutine is highlighted, the automated process is suspended for the current window. When it is determined that the code for the subroutine is not in the present graphical presentation window, the new window is automatically presented with the proper code, and then the user return to manually stepping through the code (column 5, lines 12 – 55).

In regard to claim 12, incorporating the rejection of claim 11:

> *"...wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated."*

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 13, incorporating the rejection of claim 12:

> *"...wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the line of source code in a visually distinctive manner."*

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 14, incorporating the rejection of claim 11:

> *"...further comprising the step of compiling the line before determining whether the line is represented in the graphical representation."*

The debug program of Pazel is capable of disassembling executable code that has been previously compiled (column 3, lines 60 – 66).

In regard to claim 111, incorporating the rejection of claim 11:

> *"...wherein the textual representation of the source code is obtained from the source code directly."*
>
> See Pazel column 5, lines 42 – 49.

Claims 50 – 53, and 118 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 11 – 14, and 111 (corresponding methods).

Claims 79 – 82, and 123 (data processing system) are rejected for the same corresponding reasons put forth in the rejection of claims 11 – 14, and 111 (corresponding methods).

In regard to claim 17:

"...displaying simultaneously a graphical and a textual representation of the source code;

wherein the graphical and the textual representations of the source code is generated from a language-neutral representation of the source code stored in a non-repository transient meta model, and

initiating an automated process to be performed on each of the lines of the source code;

See Pazel Figures 6 and 7 with the corresponding text for simultaneously displaying graphical and textual representation of code. Pazel does not explicitly disclose textual representation generated from a language-neutral representation stored in a non-repository transient meta model. However, Zgarba discloses the textual representation generated from a language-neutral representation stored in a non-repository transient meta model (column 5, lines 1 – 45). The model need not be in a database (a repository) format as described at column 5,

lines 18 - 19. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the simultaneous display of text and graphical of source code as taught by Pazel with the generation of graphical and text information from a non-repository transient meta model as taught by Zbarba because one would be motivated to generate updated code having immediate changes to specific parts of the software while the remaining code is untouched as taught by Zgarba in the Abstract.

> *receiving an indication to suspend the automated process when the automated process encounters a selected one of the lines; and*
>
> *while the automated process is being performed on each of the lines of source code, determining whether the line is the selected line; and*
>
> *when it is determined that the line is the selected line, suspending the automated process."*

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39). Figures 3 and 6 indicate that at least one of the lines is not represented in the graphical representation. Pazel discloses a debug action in Figure 3. When it is determined that a subroutine is highlighted, the automated process is suspended for the current window. When it is determined that the code for the subroutine is not in the present graphical presentation window, the new window is automatically presented with the proper code, and then the user return to manually stepping through the code (column 5, lines 12 – 55).

In regard to claim 18, incorporating the rejection of claim 17:

> *"...wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the portion of the*

graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated."

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 19, incorporating the rejection of claim 18:

"...wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the line of source code in a visually distinctive manner."

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 20, incorporating the rejection of claim 17:

"...further comprising the step of compiling the line before determining whether the line is the selected line."

The debug program of Pazel is capable of disassembling executable code that has been previously compiled (column 3, lines 60 – 66).

In regard to claim 117, incorporating the rejection of claim 12:

"...wherein the textual representation of the source code is obtained from the source code directly."

See Pazel column 5, lines 42 – 49.

Claims 56 – 59, and 119 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 17 – 20, and 117 (corresponding methods).

Claims 85 – 88, and 124 (data processing system) are rejected for the same corresponding reasons put forth in the rejection of claims 17 – 20, and 117 (corresponding methods).

In regard to claim 23:

"displaying simultaneously a graphical and a textual representation of the source code,

wherein the graphical and the textual representations of the source code is generated from a language-neutral representation of the source code stored in a non-repository transient meta model, and

receiving an indication of a first of the plurality of lines of the source code;

See Pazel Figures 6 and 7 with the corresponding text for simultaneously displaying graphical and textual representation of code. Pazel does not explicitly disclose textual representation generated from a language-neutral representation stored in a non-repository transient meta model. However, Zgarba discloses the textual representation generated from a language-neutral representation stored in a non-repository transient meta model (column 5, lines 1 – 45). The model need not be in a database (a repository) format as described at column 5, lines 18 - 19. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the simultaneous display of text and graphical of source code as

taught by Pazel with the generation of graphical and text information from a non-repository transient meta model as taught by Zbarba because one would be motivated to generate updated code having immediate changes to specific parts of the software while the remaining code is untouched as taught by Zgarba in the Abstract.

> *selecting a second of the plurality of lines of the source code;*
>
> *determining whether the second line is the same as the first line; and*
>
> *when it is determined that the second line is not the same as the first line, displaying the graphical representation of the second line in a visually distinctive manner."*

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 24, incorporating the rejection of claim 23:

> *"...wherein when it is determined that the second line is not the same as the first line, the method further comprises the step of displaying the second line of the source code in a visually distinctive manner."*

When the lines are scrolled, the highlight moves from one line to the next to distinctively provide a visual distinction (column 6, lines 26 – 39).

In regard to claim 25, incorporating the rejection of claim 23:

> *"...wherein when it is determined that the second line is not the same as the first line, the method further comprises the steps of:*
>
> *selecting a third of the plurality of lines of the source code;*
>
> *determining whether the third line is the same as the first line; and*

> *when it is determined that the third line is not the same as the first line, displaying the graphical representation of the third line in a visually distinctive manner."*

When the lines are scrolled, the highlight moves from one line to the next to distinctively provide a visual distinction (column 6, lines 26 – 39).

In regard to claim 26, incorporating the rejection of claim 25:

> *"...wherein when it is determined that the third line is not the same as the first line, the method further comprises the step of displaying the third line of the source code in a visually distinctive manner."*

When the lines are scrolled, the highlight moves from one line to the next to distinctively provide a visual distinction (column 6, lines 26 – 39).

In regard to claim 113, incorporating the rejection of claim 23

> *"...wherein the textual representation of the source code is obtained from the source code directly."*

See Pazel column 5, lines 42 – 49.

Claims 62 – 65, and 120 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 23 – 26, and 113 (corresponding methods).

Claims 91 – 94, and 125 (data processing system) are rejected for the same corresponding reasons put forth in the rejection of claims 23 – 26, and 113 (corresponding methods).

In regard to claim 29:

> *"...displaying a graphical representation of the plurality of lines such that at least one of the lines is not represented in the graphical representation;*
>
> *initiating an automated process on each of the lines of the source code;"*

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39). Figures 3 and 6 indicate that at least one of the lines is not represented in the graphical representation.

> *"while the automated process is being performed on each of the lines of source code, compiling the line;*
>
> *determining whether the compiled line produces an error; and*
>
> *when it is determined that the compiled line produces the error, suspending the automated process."*

Pazel discloses a debugger system that operates on compiled code, determines if an error occurs and then suspends the process (checking for breakpoints or errors) for the debugging to determine where the crash occurs (column 3, line 60 to column 4, line 12).

In regard to claim 30, incorporating the rejection of claim 29:

> *"...wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated."*

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, inherently giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 31, incorporating the rejection of claim 30:

> *"...wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the line of source code in a visually distinctive manner."*

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 114, incorporating the rejection of claim 29

> *"...wherein the textual representation of the source code is obtained from the source code directly."*
>
> See Pazel column 5, lines 42 – 49.

Claims 68 – 70, and 121 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 29 – 31, and 114 (corresponding methods).

Claims 97 – 99, and 126 (data processing system) are rejected for the same corresponding reasons put forth in the rejection of claims 29 – 31, and 114 (corresponding methods).

In regard to claim 34:

"..."displaying simultaneously a graphical and a textual representation of the source code,

wherein the graphical and the textual representations of the source code is generated from a language-neutral representation of the source code stored in a non-repository transient meta model; and

See Pazel Figures 6 and 7 with the corresponding text for simultaneously displaying graphical and textual representation of code. Pazel does not explicitly disclose textual representation generated from a language-neutral representation stored in a non-repository transient meta model. However, Zgarba discloses the textual representation generated from a language-neutral representation stored in a non-repository transient meta model (column 5, lines 1 – 45). The model need not be in a database (a repository) format as described at column 5, lines 18 - 19. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the simultaneous display of text and graphical of source code as taught by Pazel with the generation of graphical and text information from a non-repository transient meta model as taught by Zbarba because one would be motivated to generate updated code having immediate changes to specific parts of the software while the remaining code is untouched as taught by Zgarba in the Abstract.

selecting one of the plurality of lines of the source code;

compiling the selected line;

determining whether the compiled line produces an error; and

when it is determined that the compiled line does not produce an error, displaying the graphical representation of the selected line in a visually distinctive manner."

Pazel discloses a debugger system that operates on compiled code, determines if an error occurs and then suspends the process (checking for breakpoints or errors) for the debugging to determine where the crash occurs (column 3, line 60 to column 4, line 12).

In regard to claim 35, incorporating the rejection of claim 34:

"...wherein when it is determined that the compiled line does not produce an error, the method further comprises the step of displaying the selected line of source code in a visually distinctive manner."

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 36, incorporating the rejection of claim 34:

"...wherein when it is determined that the compiled line does not produce an error, the method further comprises the steps of:

selecting a second of the plurality of lines of the source code;

compiling the second line;

determining whether the compiled second line produces an error; and

when it is determined that the compiled second line does not produce an error, displaying the graphical representation of the second line in a visually distinctive manner."

Pazel discloses an automated process that processes the lines and displays them in a visually distinctive way by highlighting the currently active line after scrolling the lines, line by line successively, giving an animated appearance (column 6, lines 26 – 39).

In regard to claim 37, incorporating the rejection of claim 36:

> *"...wherein when it is determined that the compiled second line does not produce an error, the method further comprises the step of displaying the second line of source code in a visually distinctive manner."*

When the lines are scrolled, the highlight moves from one line to the next to distinctively provide a visual distinction (column 6, lines 26 – 39).

In regard to claim 115, incorporating the rejection of claim 34:

> *"...wherein the textual representation of the source code is obtained from the source code directly."*

See Pazel column 5, lines 42 – 49.

Claims 73 – 76, and 122 (computer readable medium) are rejected for the same corresponding reasons put forth in the rejection of claims 34 – 37, and 115 (corresponding methods).

Claims 102 – 105, and 127 (data processing system) are rejected for the same corresponding reasons put forth in the rejection of claims 34 – 37, and 115 (corresponding methods).

6. Claims 4, 5; 9, 10; 15, 16; 21, 22; 27, 28; 32, 33; 38, 39; 43, 44; 48, 49; 54, 55; 60, 61; 66, 67; 71, 72; 77, 78; 83, 84; 89, 90; 95, 96; 100, 101; 106, and 107 are rejected under 35 U.S.C.

103(a) as being unpatentable over Pazel, U.S. Patent 5,410,648 (Reissue 36,422) in view of Zgarba et al, U.S. Patent 6,502,239, and further in view of Graham, U.S. Patent 5,918,053.

In regard to claim 4, incorporating the rejection of claim 1:

"...wherein the graphical representation comprises a class diagram."

Pazel discloses a flow graph, but neither Pazel nor Zgarba explicitly discloses a class diagram. However, Graham discloses class diagrams derived from underlying code (e.g., Figures 15 – 17). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the flow graph capability in the Pazel invention with the well known diagramming of classes as taught by Graham, because the combination logically follows if one is debugging an object-oriented application with the display capability of program flow in Pazel.

In regard to claim 5, incorporating the rejection of claim 1:

"...wherein the graphical representation comprises a sequence diagram."

Pazel discloses a flow graph, but neither Pazel nor Zgarba explicitly discloses a sequence diagram. However, Graham discloses sequence diagrams (e.g., Figures 7 – 14). Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the flow graph capability in the Pazel invention with the well known diagramming of steps in a sequence as taught by Graham, because the combination logically follows if one is debugging an object-oriented application with the display capability of program flow in Pazel adding an analytical feature that does not require the code to be completely executed, as taught by Graham at column 7, lines 15 – 17, to determine object interaction.

In regard to claim 9, incorporating the rejection of claim 6:

"...wherein the graphical representation comprises a class diagram."

Rejected for the same reasons put forth in the rejection of claim 4 above.

In regard to claim 10, incorporating the rejection of claim 6:

"...wherein the graphical representation comprises a sequence diagram."

Rejected for the same reasons put forth in the rejection of claim 5 above.

In regard to claim 15, incorporating the rejection of claim 11:

"...wherein the graphical representation comprises a class diagram."

Rejected for the same reasons put forth in the rejection of claim 4 above.

In regard to claim 16, incorporating the rejection of claim 11:

"...wherein the graphical representation comprises a sequence diagram."

Rejected for the same reasons put forth in the rejection of claim 5 above.

In regard to claim 21, incorporating the rejection of claim 17:

"...wherein the graphical representation comprises a class diagram."

Rejected for the same reasons put forth in the rejection of claim 4 above.

In regard to claim 22, incorporating the rejection of claim 17:

"...wherein the graphical representation comprises a sequence diagram."

Rejected for the same reasons put forth in the rejection of claim 5 above.

In regard to claim 27, incorporating the rejection of claim 23:

"...wherein the graphical representation comprises a class diagram."

Rejected for the same reasons put forth in the rejection of claim 4 above.

In regard to claim 28, incorporating the rejection of claim 23:

"...wherein the graphical representation comprises a sequence diagram."

Rejected for the same reasons put forth in the rejection of claim 5 above.

In regard to claim 32, incorporating the rejection of claim 29:

"...wherein the graphical representation comprises a class diagram."

Rejected for the same reasons put forth in the rejection of claim 4 above.

In regard to claim 33, incorporating the rejection of claim 29:

"...wherein the graphical representation comprises a sequence diagram."

Rejected for the same reasons put forth in the rejection of claim 5 above.

In regard to claim 38, incorporating the rejection of claim 34:

"...wherein the graphical representation comprises a class diagram."

Rejected for the same reasons put forth in the rejection of claim 4 above.

In regard to claim 39, incorporating the rejection of claim 34:

"...wherein the graphical representation comprises a sequence diagram."

Rejected for the same reasons put forth in the rejection of claim 5 above.

Response to Arguments

7. Applicant's arguments with respect to claims 1 - 128 have been considered but are moot in view of the new ground(s) of rejection.

However, the following point of interpretation will be made:

The Applicant has argued:

"It is true that the Pazel patent does describe a similar feature that highlights the active line of code while performing the debugger program. However, unlike the present invention, the Pazel program does not perform this feature in the context of a software development tool in the same textual and graphical views used for the manipulation of the software during the development process."

Examiner's response:

In a broad sense a debugger is a software development tool. Further, the claim language does not exclude the debugger interpreted in this broad sense. The context presented in the claims is a data processing system, which is not limited to simply a software development tool. Therefore, the Pazel invention is maintained in the rejection in view of this interpretation.

Conclusion

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lawrence Shrader whose telephone number is (571) 272-3734. The examiner can normally be reached on M-F 08:00-16:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Lawrence Shrader
Examiner
Art Unit 2124

26 January 2005

Kakali Chaki

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100